Ethernet controller **140** along with an active running DMA channel command triggers a DMA transfer.

A process for transmitting data packets from the system memory to an external device such as the Ethernet controller **24** will now be described with reference to FIGS. **4**, **5**, and **6**. In step S1 in FIG. **5**, the memory access controller, which in the example is I/O controller **10**, transfers a data unit in the form of one or more data packets from the memory **90** (see FIG. **1**) to an I/O device, labelled an external device controller, e.g., Ethernet controller **140**. This is also shown graphically in FIG. **6**. The external device controller **140** then sends the data unit to the external system **150** Specifically, the memory access controller **10** transfers the data unit to the bus interface **180** which transfers the data packets to the transmit FIFO buffer **170**. The I/O device controller **140** then transmits the data unit to the external system **150**. In step S2, the I/O device controller **140** waits to receive an indication from the external system **150** that the data unit has been successfully transmitted. After the I/O device controller **140** has received the indication from the external system **150**, in step S3, it sends an interrupt signal over interrupt line **240** to the memory access controller **10**. In the specific embodiment depicted, the media access controller **200** sets its status registers **250** to a setting that indicates that the complete packet was sent.

The interrupt signal causes a bit to be set in the memory access controller **10**'s channel status register **110** which, in the example shown in FIG. **3**, is bit s5, thus storing an indication of receipt of the interrupt signal. This information can then be used to control subsequent operation of the memory access controller **10**. For example, the memory access controller can be in a wait condition until the status bit is set, which will cause the memory access controller to execute a next instruction. Also, as shown in step S4, it can be used to prompt the memory access controller **10** to request status information from the status registers **250**, which the external device controller then provides. The process can be repeated to transmit another packet before the channel goes idle. Assuming normal operation, the system processor does not need to be interrupted until the last packet has been sent.

The Ethernet controller **140** will also produce an interrupt signal if an error occurs during transmission.

One example of how the bits s7–s0 can be arranged in the channel status register **110** is illustrated in FIG. **7**, wherein only the bits s7, s5, and sO may be asserted to generate actions by the DMA channel. In this example, the DMA controller **60** interprets the set status bit to means that the packet was transmitted by the Ethernet controller **140**. As a result, the I/O controller **10** reads the status information stored in the status registers **250**. It will be appreciated by one skilled in the art that the interrupt signal can also be transmitted only after the completion of the last data packet in a data stream.

The operation of an exemplary Ethernet receive channel is illustrated in FIG. **8**. In step S1', data packets are being written into a system memory **90** from an external device **150**. When the Ethernet controller **140** determines that data packet transmission to the I/O controller **10** has been completed (step S2'), the Ethernet controller **140** sets the status register **250** to indicate the status of the packet that was just transmitted and generates an interrupt signal as a status bit in step S3'. One example of how the status bits can be arranged for the receive channel status register is illustrated in FIG. **9**. In this embodiment of the present invention, the status bit s6 indicates that the packet has been sent. By tying the interrupt signal to the s6 bit of the receive channel status register, the interrupt signal will indicate to the I/O controller **10** that the end of the packet has been received.

In response to receipt of the interrupt/status bit, the I/O controller **10** requests the status information contained in the status registers **250** in the Ethernet controller **140** in step S4'. The Ethernet controller **140** then in step S5' sends the status information (which, for example, may be four bytes) to the I/O controller **10** which writes the status information on to the end of the received packet in step S6'. Thus, the status bit s6 can be used to indicate the end of a packet, which in turn means that the buffer size can be made smaller than the maximum data packet size. This allows for a more efficient use of the available buffer space since the buffers are no longer tied to a one-to-one correspondence with the data packets.

It will be appreciated by those skilled in the art that the present invention can be embodied in other specific forms without departing from the spirit or essential character thereof. The presently disclosed embodiments are therefore considered in all respects to be illustrative and not restrictive. The scope of the invention is indicated by the appended claims rather than the foregoing description, and all changes which come within the meaning and range of equivalents thereof are intended to be embraced herein.

I claim:

1. A method of transferring a data unit from a computer system memory and to an external system through an I/O device using a memory access controller, said memory access controller including a register for storing information which the memory access controller uses to control its own operation, said method comprising the steps of:

a first step, executed by said memory access controller, of retrieving said data unit from said computer system memory and transmitting said data unit to said I/O device;

a second step, executed by said I/O device, of transmitting said data unit retrieved and transmitted in said first step to said external system;

a third step, executed by said I/O device, of sending a data status signal to said memory access controller when said second step is complete; and

a fourth step, executed by said memory access controller, of storing an indication of said data status signal sent in said third step in said register.

2. A method as in claim **1** wherein said data status signal indicates a successful transmission of said data unit to said external system by said I/O device, wherein said register is a channel status register, and wherein a bit of said channel status register assumes a value indicative of receipt of said data status signal.

3. A method as claimed in claim **1** wherein said memory access controller waits until said bit reflects a value of said data status signal indicating successful transmission of a data unit to the external system before sending another data unit.

4. A method as claimed in claim **1** wherein said external system is a network, said I/O device is an Ethernet controller, and said data unit is a packet.

5. A method as claimed in claim **1** wherein said data status signal is an interrupt signal.

6. A method as claimed in claim **1** wherein said memory access controller is a DMA channel controller.

7. A method as claimed in claim **1** wherein there are two DMA channels, a receive channel and a transmit channel, and wherein an interrupt from a said receive channel is masked.

8. A computer system comprising:

a computer system memory;

an I/O device connected to an external system to transfer data units between said computer system memory and an external system; said I/O device including a data status signal generator which generates a data status signal upon completion of transfer of a data unit;

a memory access controller connected to said computer system memory and said I/O device; said memory access controller including a register for storing status information which the memory access controller uses to control its own operation, wherein said memory access controller receives said data status signal and stores an indication of a value of said data status signal in said register, wherein during a data transmit operation the memory access controller retrieves said data unit from said computer system memory and transfers said data unit to said I/O device and said I/O device transmits said data unit to said external system and wherein the said data transfer status signal is sent from the I/O device to the memory access controller after the transfer of said data unit to said I/O device.

9. A computer system as claimed in claim 8 wherein said data status signal indicates a successful transmission of a data unit to said external system by said I/O device, wherein said register is a channel status register, and wherein a bit of the channel status register is arranged to reflect a value of said data status signal.

10. A computer system as claimed in claim 9 wherein said memory access controller waits until said bit reflects a value of said data status signal indicating successful transmission of a data unit to the external system before sending another data unit.

* * * * *